(14) Technical Report
SMU-EE-TR-81-9
Contract Number N00014-79-C-0494
April 24, 1981
(15)

AD A098285

LEVEL II

(6) Note on Statistical
Texture Discrimination*

(9) Technical rept.

(11) 24 Apr 81

DTIC
ELECTE
APR 28 1981
S C D

(12) 24

by

(10) C. H. Chen
Rong-Hwang Wu
Electrical Engineering Department
Southeastern Massachusetts University
North Dartmouth, Massachusetts 02747

DTIC FILE COPY

*The support of the Statistics and Probability Program
of the Office of Naval Research on this work is
gratefully acknowledged.

407932

81 4 27 138

Note on Statistical Texture Discrimination

C. H. Chen

Rong-Hwang Wu

1. Summary

For a given textured image, experimental results on texture feature extraction, dimensionality reduction, and iterative Bayes classification are reported. The classification performance is superior to other texture discrimination methods that have been examined on the same data. While the method presented is recommended as an effective statistical texture discrimination approach, the large amount of computation required, especially in feature extraction, can be undesirable in some applications. Detailed computer program listing is given in the Appendix.

2. Texture Feature Extraction

The image examined is a 64 x 64 textured image described in an earlier report [1]. Because the image size is small, the co-occurrence matrix is computed for distance 1. The number of gray levels is 16. The 16 x 16 co-occurrence matrix P is generated [2] for angles $\theta = 0^\circ, 45^\circ, 90^\circ$, and 135°. For each P matrix, we compute the following four measurements.

(i) Angular Second Moment

$$f_1 = \sum_{i=1}^{N-1} \sum_{j=1}^{N-1} (P_{ij})^2$$

where N is the number of gray levels and P_{ij} is an element of matrix P.

(ii) Inertia or Contrast

$$f_2 = \sum_{i=1}^{N-1} \sum_{j=1}^{N-1} (i-j)^2 P_{ij}$$

(iii) Correlation

$$f_3 = \frac{1}{\sigma_i \sigma_j} \left[\sum_{i=1}^{N-1} \sum_{j=1}^{N-1} (ij) P_{ij} - \hat{\mu}_i \hat{\mu}_j \right]$$

where $$\hat{\mu}_i = \sum_{i=o}^{N-1} \sum_{j=o}^{N-1} i\, P_{ij}$$

$$\hat{\mu}_j = \sum_{i=o}^{N-1} \sum_{j=o}^{N-1} j\, P_{ij}$$

$$\sigma_i^2 = \sum_{i=o}^{N-1} \sum_{j=o}^{N-1} P_{ij}(i - \hat{\mu}_i)^2$$

$$\sigma_j^2 = \sum_{i=0}^{N-1} \sum_{j=o}^{N-1} P_{ij}(j - \hat{\mu}_j)^2$$

(iv) $$f_4 = \sum_{i=0}^{N-1} \sum_{j=o}^{N-1} |i - j|^3 \log_{10}(P_{ij} + 1)$$

Despite the directional effect, we take the average and the maximum difference of the functional values, as they vary with θ. A set of 8 texture features is formed as

$$x = [\bar{f}_1, \Delta f_1, \bar{f}_2, \Delta f_2, \bar{f}_3, \Delta f_3, \bar{f}_4, \Delta f_4]^T$$

where $\bar{f}$ = the value averaged over the four directions $\theta = 0^\circ, 45^\circ, 90^\circ$ and 135°

$$\Delta f_i = f_{i\,max} - f_{i\,min}$$

3. Feature Space Transformation

We now consider the problems of finding the optimum discriminat vectors and transforming the original feature space to the maximum discriminant feature space. After the texture features are generated, we choose a set of 25 x 25 learning feature sets for each class and follow the method developed by Foley and Sammon [3] to calculate the first three largest discriminant values (γ_1, γ_2, & γ_3) and their corresponding vectors (d_1, d_2, & d_3). By using these three vectors we can transform x from original 8-dimensional feature space into a 3-dimensional maximum discriminant feature space by

$$y = Ax$$

where $A = [d_1, d_2, d_3]^T$ is a 3 x 8 matrix

x is an 8 x 1 feature vector

y is a 3 x 1 feature vector

4. Texture classification

The use of k-NN decision rule suggested by Stark and O'Toole [4] provided poor result while requiring an enormous computer time for large number of learning samples. An iterative Bayes decision rule as proposed in the following is much more effective. A multivariate Gaussian assumption is made for each class with mean vectors M_i and covariance matrices Σ_i, $i = 1,2$. Also let $P(\omega_i)$ be the a priori probability. The Bayes classification rule is therefore

$$\frac{1}{2}(y - M_1)^T \Sigma_1^{-1} (y - M_1) - \frac{1}{2}(y - M_2)^T \Sigma_2^{-1} (y - M_2)$$

$$+ \frac{1}{2} \ln \frac{|\Sigma_1|}{|\Sigma_2|} \lessgtr \ln \frac{P(\omega_1)}{P(\omega_2)} \rightarrow y \varepsilon \begin{matrix} \omega_1 \\ \omega_2 \end{matrix}$$

for pixel y in the transformed space. Initially we assume $P(\omega_1) = P(\omega_2) = 0.5$. After the decision is made for the pixel, and the pixels of its 5 x 5 neighborhood, the percentage number of pixels within the 5 x 5 neighborhood classified as ω_i is used as updated value of $P(\omega_i)$. The procedure is performed iteratively to update $P(\omega_i)$ in each iteration.

5. Computer Results

Figure 1 is the ideal segmentation of the textured image [1]. Figure 2 is the result of using the Bayes decision rule with equal a priori probability (without any iteration). Figure 3 is the result of one iteration while Fig. 4 is the result of 4 iterations. The convergence is extremely fast as it takes only 4 iterations to provide a low error rate of 1.5625%. In the previous work [1], the use of maximum a posteriori estimation has an error rate of 3.9C%. More recently we have reported [5] an error rate of 2.05% by using Fisher's linear discriminant and the texture features of angular second moment, constrast, and entropy. Thus the present method has the lowest error rate. It does require, however, more than twice of the

computation time as compared with the Fisher's linear discriminant method. Computation of texture features is most time consuming. In the future when the co-occurrence matrix computer is developed, the computation time can be greatly reduced.

References

1. C.H. Chen, R.H. Wu and C. Yen, "Some experimental results on linear estimation for image analysis", Technical Report SMU-EE-TR-81-4, January 1981.

2. W.K. Pratt, "Digital Image Processing", Wiley, 1978.

3. D.H. Foley and J.W. Sammon, "An optimal set of discriminant vectors", IEEE Trans. on Computers, Vol. C-24, pp. 281-289, March 1975.

4. R.K. O'Toole and H. Stark, "A comparative study of texture discrimination using an optical/digital computer versus all-digital methods", in "Pattern Recognition in Practice", edited by E. Gelsema and L.N. Kanal, North-Holland Publishing Co., 1980.

5. C.H. Chen, "On the statistical image segmentation techniques", to appear in Proc. of IEEE Pattern Recognition and Image Processing Conference, August 1981.

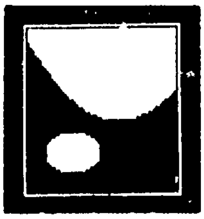

Figure 1

Figure 2

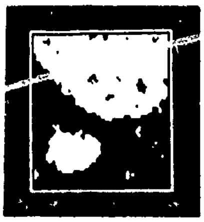

Figure 3

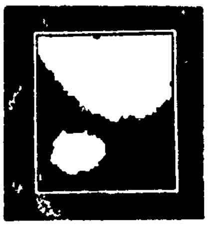

Figure 4

Appendix

Computer Program Listing

Note: Feature extraction is performed by File [50,50]

Feature space transformation is performed by File CHC [50,50]
File ODV1 [50,50]   and File COMP [50,50]

Classification is performed by File CL2 [50,50]   File PRIK [50,50]

```
C
C       FILE: FEAC[50,50]
C
C       *************************************************
C
C       FEATURE EXTRACTION
C
C       *************************************************
        DIMENSION A(8,64),T(64),IT(64)
        COMMON NF,NQ,NQ1,LL,IDA(64,64),P(16,16),I1,J1,L1,QM(8),DF(16)
        WRITE(6,111)
111     FORMAT(1X,'WHICH FEATUR THAT YOU WANT TO CHANGE ?'/
        11X,'FIRST FEATURE,NF=1'/1X,'SECOND FEATURE,NF=2'/
        21X,'THREE FEATURE,NF=3'/1X,'FOURTH FEATURE,NF=4'/
        31X,'NF=?'/'0')
        READ(6,112)NF
112     FORMAT(I4)
        WRITE(6,113)
113     FORMAT(1X,'CHOOSE A NEW FEATURE'/
        11X,'ANGULAR SECOND MOMANTENT, NC=1'/
        21X,'INERTIA,NC=2'/1X,'AUTOCORRELATION NC=3'/
        31X,'ENTROPY,NC=4'/1X,'JOINT PROBABILITY,NC=5'/
        41X,'MIXED,NC=6'/1X,'NC= ?'/'0')
        READ(6,112)NC
        WRITE(6,114)NF,NC
114     FORMAT(5X,'NF=',I5,2X,'NC=',I5/)
        NQ1=NF*2
        NQ=NQ1-1
C       *************************************************
C
C       CHANGE ORIGINAL 256 GRAY LEVEL TO 16 GRAY LEVELS
C
C       *************************************************
        DEFINE FILE 3(64,64,U,INDEX)
        INDEX=1
        DO 100 I=1,64
        READ(3'INDEX)IT
        DO 100 J=1,64
        DO 100 K=1,16
        IL=(K-1)*15
        IH=K*15
        IF(IT(J).LT.IL.OR.IT(J).GE.IH)GO TO 100
        IDA(I,J)=K
100     CONTINUE
        DO 101 IA=1,64
        WRITE(6,102)(IDA(IA,IB),IB=1,16)
102     FORMAT(1X,16I2)
101     CONTINUE
        END FILE 3
C       ************************************************************
C
C       FILE 2 LINE 1 TO 512 ARE STORED ORIGINAL FEATURES,
C       EACH PIXEL HAS 8 FEATURE.
C       LINE 521 TO 712 ARE STORED COMPRESSED FEATURES,
C       EACH PIXEL HAS MAX. 3 DIMENSIONS
C       LINE 721 TO 784 AND LINE 801 TO 864 ARE STORED TWO 64*64
C       IMAGES
C
C       ************************************************************
```

```
      DEFINE FILE 2(864,128,U,INDEX)
      DO 1 I=2,63
      I1=I
      DO 2 J=2,63
      J1=J
      DO 31 K=1,8
31    QM(K)=0.
      DO 33 K1=1,16
33    DF(K1)=0.
      DO 4 L=1,4
      L1=L
      LL=(NF-1)*4+L1
      DO 32 KA=1,16
      DO 32 KB=1,16
32    P(KA,KB)=0.
      CALL GDM
      CALL JOIN(NC)
4     CONTINUE
      CALL MAXDF
      QM(NQ)=QM(NQ)/4.
      DO 6 N=NQ,NQ1
6     A(N,J)=QM(N)
2     CONTINUE
11    FORMAT(//60X,'I(',I3,1X,')'//8(F15.5,5X))
      DO 7 NS=NQ,NQ1
      INDEX=(I-1)*8+NS
      DO 8 NT=1,64
8     T(NT)=A(NS,NT)
      WRITE(2'INDEX)T
7     CONTINUE
      WRITE(6,10)I
10    FORMAT(10X,I5)
1     CONTINUE
      CALL BELL
      CALL EXIT
      END
C     ****************************************************
C
C     SPATIAL-DEPENDENCE PROBABILITY DISTRIBUTION.
C
C     ****************************************************
      SUBROUTINE GDM
      COMMON NF,NQ,NQ1,LL,IDA(64,64),P(16,16),I1,J1,L1,QM(8),DF(16)
      C=0.
      IS=I1-2
      DO 1 I=1,3
      IS=IS+1
      JS=J1-2
      DO 2 J=1,3
      JS=JS+1
      GOTO(11,12,13,14)L1
11    IT=IS
      JT=JS+1
      GO TO 15
12    IT=IS-1
      JT=JS+1
      GO TO 15
13    IT=IS-1
      JT=JS
      GO TO 15
14    IT=IS-1
      JT=JS-1
```

```
15      ID=IABS(IT-I1)
        JD=IABS(JT-J1)
        IF(ID.GT.1.OR.JD.GT.1)GO TO 2
        M=IDA(IS,JS)
        N=IDA(IT,JT)
        P(M,N)=P(M,N)+1.
        P(N,M)=P(N,M)+1.
        C=C+2.
2       CONTINUE
1       CONTINUE
        DO 6 K=1,16
        DO 6 L=1,16
6       P(K,L)=P(K,L)/C
        RETURN
        END
C       ************************************************
C
C       101 : ANGULAR SECOND MOMENT
C       102 . INERTIA
C       103 : CORRELATION
C       104 : ENTROPY
C       105 : JOIN PROBABILITY
C       106 : ((I-J)**3)*LOG(P(I,J)+1.)
C
C       ************************************************
        SUBROUTINE JOIN(NC)
        COMMON NF,NQ,NQ1,LL,IDA(64,64),P(16,16),I1,J1,L1,QM(8),DF(16)
        GOTO(101,102,103,104,105,106),NC
101     F1=0.
        DO 11 I=1,16
        DO 11 J=1,16
        IF(P(I,J).EQ.0.)GO TO 11
        F1=F1+P(I,J)*P(I,J)
11      CONTINUE
        QM(NQ)=QM(NQ)+F1
        DF(LL)=F1
        GO TO 100
102     F2=0.
        DO 21 I=1,16
        II=I-1
        DO 21 J=1,16
        IF(P(I,J).EQ.0 )GO TO 21
        JJ=J-1
        DIJ=FLOAT(II-JJ)
        D2=DIJ*DIJ
        F2=F2+D2*P(I,J)
21      CONTINUE
        QM(NQ)=QM(NQ)+F2
        DF(LL)=F2
        GO TO 100
103     F3=0.
        UX=0.
        UY=0.
        QXS=0.
        QYS=0.
        DO 31 I=1,16
        CI=FLOAT(I-1)
        DO 31 J=1,16
        IF(P(I,J).EQ.0.)GO TO 31
        CJ=FLOAT(J-1)
        UX=UX+CI*P(I,J)
        UY=UY+CJ*P(I,J)
```

```
31        CONTINUE
          DO 32 I=1,16
          CI=FLOAT(I-1)
          DO 32 J=1,16
          IF(P(I,J).EQ.0.)GO TO 32
          CJ=FLOAT(J-1)
          FC1=CI*CI*P(I,J)
          FC2=CJ*CJ*P(I,J)
          QXS=QXS+FC1
          QYS=QYS+FC2
32        CONTINUE
          QXS=QXS-UX*UX
          QYS=QYS-UY*UY
          IF(QXS.EQ.0..OR.QYS.EQ.0.)GO TO 10
          UXY=UX*UY
          QX=SQRT(QXS)
          QY=SQRT(QYS)
          QXY=QX*QY
          DO 33 I=1,16
          CI=FLOAT(I-1)
          DO 33 J=1,16
          CJ=FLOAT(J-1)
          IF(P(I,J).EQ.0.)GO TO 33
          F3=F3+(CI*CJ*P(I,J))
33        CONTINUE
          F3=(F3-UXY)/QXY
          QM(NQ)=QM(NQ)+F3
          GO TO 17
10        F3=1.0
          QM(NQ)=QM(NQ)+1.0
          WRITE(6,12)I1,J1,L1
12        FORMAT(1X,'I1=',I3,2X,'J1=',I3,2X,'L1=',I3,2X,
          1'VARIANCE = 0'//)
17        DF(LL)=F3
          GO TO 100
104       F4=0.
          DO 41 I=1,16
          DO 41 J=1,16
          IF(P(I,J).EQ.0.)GO TO 41
          F4=F4+P(I,J)*ALOG10(P(I,J))
41        CONTINUE
          QM(NQ)=QM(NQ)+F4
          DF(LL)=F4
          GO TO 100
105       F5=0.
          DO 51 I=1,16
          II=I-1
          DO 51 J=1,16
          IF(P(I,J).EQ.0.)GO TO 51
          JJ=J-1
          FIJ=FLOAT(II*JJ)
          F5=F5+FIJ*P(I,J)
51        CONTINUE
          QM(NQ)=QM(NQ)-F5
          DF(LL)=F5
          GO TO 100
106       F6=0.
          DO 61 I=1,16
          II=I-1
          DO 61 J=1,16
          IF(P(I,J).EQ.0.)GO TO 61
          JJ=J-1
          VIJ=FLOAT(II-JJ)
```

```
      FIJ=ABS(VIJ)
      VV=FIJ*FIJ
      V=VV*FIJ
      F6=F6+V*ALOG10(P(I,J)+1.)
61    CONTINUE
      QM(NQ)=QM(NQ)+F6
      DF(LL)=F6
100   RETURN
      END
      SUBROUTINE MAXDF
      COMMON NF,NQ,NQ1,LL,IDA(64,64),P(16,16),I1,J1,L1,QM(8),DF(16)
      REAL D(6),DC(4)
      II=(NF-1)*4
      IM=NF*2
      M=0
      DO 2 J=1,4
2     DC(J)=DF(II+J)
      DO 7 K=1,3
      KS=K+1
      DO 7 KK=KS,4
      M=M+1
7     D(M)=ABS(DC(K)-DC(KK))
      DM=D(1)
      DO 3 L=2,6
      IF(DM.GT.D(L))GO TO 3
      DM=D(L)
3     CONTINUE
      QM(IM)=DM
      RETURN
      END
C     ****************************************
C     FILE:  CHC [50,50]
C     CHOOSE LEARNING SAMPLES
C
C     ****************************************
      REAL A(200,25),F(64),R(25)
      DEFINE FILE 2(864,128,U,INDEX)
      WRITE(6,10)
10    FORMAT(1X,'READ NF,IXI,IYI,IXF,IYF'/'0')
      READ(6,11)NF,IXI,IYI,IXF,IYF
11    FORMAT(5I5)
      IS=(IYI-1)*8+1
      IT=IYF*8
      ID=IT-IS+1
      INDEX=IS
      IX=IXF-IXI+1
      IX1=IXI-1
      DO 1 I=1,ID
      READ(2'INDEX)F
      DO 1 J=1,IX
1     A(I,J)=F(IX1+J)
      END FILE 2
      DEFINE FILE 1(408,50,U,INDEX)
      INDEX=(NF-1)*200+1
      DO 2 K=1,200
      DO 3 L=1,25
3     R(L)=A(K,L)
      WRITE(1'INDEX)R
2     CONTINUE
      CALL BELL
      CALL EXIT
      END
```

```
C
C       FILE: ODV1[50,50]
C
C       ************************************************
C
C       FIND A OPTIMAL SET OF DISCRIMINANT VECTORS
C
C       ************************************************
        REAL A(25),U(2,8),DT(8),S(8,8)
        REAL SI(8,8),D(8,25),WW(2,8,8),W(8,8)
        REAL Y(8),WI(8,8),WS(8,8),T(8),OD(8),ODV(8,5)
        REAL W1(8,8),W2(8,8),CL(6),SN1(6),DSC(8)
        DEFINE FILE 1(408,50,U,INDEX)
        WRITE(6,10)
10      FORMAT(1X,'SIZE OF LEARNING SAMPLES (N*N)'/'0')
        READ(6,11)N
11      FORMAT(I3)
C       ************************************************
C
C       FIND MEAN U1 & U2 AND DIFFRENCE DT
C
C       ************************************************
        DO 1 I=1,2
        I1=(I-1)*200
        DO 2 J=1,N
        J1=(J-1)*8
        DO 3 K=1,8
        INDEX=I1+J1+K
        READ(1'INDEX)A
        DO 3 L=1,N
        U(I,K)=U(I,K)+A(L)
3       CONTINUE
2       CONTINUE
        DO 4 M=1,8
4       U(I,M)=U(I,M)/FLOAT(N*N)
1       CONTINUE
        DO 12 ID=1,2
        WRITE(5,33)ID,(U(ID,K),K=1,8)
33      FORMAT(1X,'U(',I2,1X,')=',8F15.5/)
12      CONTINUE
        DO 5 I=1,8
5       DT(I)=U(1,I)-U(2,I)
        WRITE(5,34)DT
34      FORMAT(//1X,'DT=',8F15.5)
C       ****************************************************
C
C       FIND WITHIN-CLASS SCATER MATRIX A AND ITS INVERSE MATIX
C
C       ****************************************************
        DO 6 I=1,2
        I1=(I-1)*200
        DO 7 J=1,N
        J1=(J-1)*8
        INDEX=I1+J1+1
        DO 8 K=1,8
        READ(1'INDEX)A
        DO 8 L=1,25
        D(K,L)=A(L)
8       CONTINUE
```

```
      DO 9 N1=1,N
      DO 9 K1=1,8
      DO 9 L1=1,8
      DD1=D(K1,N1)-U(I,K1)
      DD2=D(L1,N1)-U(I,L1)
      WW(I,K1,L1)=WW(I,K1,L1)+DD1*DD2
9     CONTINUE
7     CONTINUE
      DO 15 K2=1,8
      DO 15 L2=1,8
      WW(I,K2,L2)=WW(I,K2,L2)/FLOAT(N*N)
15    CONTINUE
6     CONTINUE
      DO 110 KS=1,8
      DO 110 KT=1,8
      W1(KS,KT)=WW(1,KS,KT)
      W2(KS,KT)=WW(2,KS,KT)
110   CONTINUE
      DO 116 I=1,8
      DO 116 J=1,8
116   W(I,J)=W1(I,J)+W2(I,J)
      WRITE(5,117)W1
      WRITE(5,118)W2
117   FORMAT(//40X,'W1'/8(F15.4))
118   FORMAT(//40X,'W2'/8(F15.4))
      WRITE(5,119)W
119   FORMAT(//40X,'W'/8(F15.4))
      ND=8
      CALL INVERS(W,W1,ND)
      WRITE(6,199)
199   FORMAT(1X,'INVERSE W OK!'/'0')
C     ***********************************************
C
C     FINF 'ALPHA 1'
C
C     ***********************************************
      DO 21 I=1,8
      DO 21 J=1,8
      SUM=0.
      DO 22 K=1,8
22    SUM=SUM+W1(I,K)*W1(K,J)
      WS(I,J)=SUM
21    CONTINUE
      AL=0.
      DO 24 J=1,8
      SUM=0.
      DO 25 K=1,8
25    SUM=SUM+WS(J,K)*DT(K)
      T(J)=SUM
24    CONTINUE
      DO 23 I=1,8
      AL=AL+DT(I)*T(I)
23    CONTINUE
      AL=1./SQRT(AL)
C     ***********************************************
C
C     FINF D1
C
C     ***********************************************
      DO 26 I=1,8
      SUM=0.
      DO 27 J=1,8
27    SUM=SUM+W1(I,J)*DT(J)
      OD(I)=SUM*AL
26    CONTINUE
```

```
C
C     FINE "DISCRIM-VALUE R1"
C
C     **********************************************
      NE=1
      CALL RNV(W,OD,DT,NE)
C     **********************************************
C
C     FINE S11 FOR S(1,1) & S-1(1,1)
C
C     **********************************************
      DO 29 J=1,8
      SUM=0.
      DO 30 K=1,8
30    SUM=SUM+WI(J,K)*OD(K)
      T(J)=SUM
29    CONTINUE
      S11=0.
      DO 28 I=1,8
28    S11=S11+OD(I)*T(I)
      S(1,1)=S11
      SI(1,1)=1./S11
C     **************************************************
C
C     SET COLOUM [1/AL 0 0 0 .....] AND STORE "D1" TO "ODV"
C
C     **************************************************
      CL(1)=1./AL
      DO 41 I=2,7
41    CL(I)=0.
      DO 148 J=1,8
148   ODV(J,1)=OD(J)
C     ************************************************
C
C     DO LOOP FINE D,R & S
C
C     ************************************************
      DO 40 ND=1,4
      NF=ND+1
      IF(ND.EQ.1)GO TO 44
C     **********************************************
C
C     FINE MATRIC S AND ITS INVERSE MATRIX
C
C     **********************************************
      DO 61 K1=1,8
      SUM=0.
      DO 62 K2=1,8
62    SUM=SUM+WI(K1,K2)*ODV(K2,ND)
      T(K1)=SUM
61    CONTINUE
      DO 63 K3=1,ND
      SUM=0.
      DO 64 K4=1,8
64    SUM=SUM+ODV(K4,K3)*T(K4)
      Y(K3)=SUM
63    CONTINUE
      DO 65 K5=1,ND
      S(K5,ND)=Y(K5)
      S(ND,K5)=Y(K5)
65    CONTINUE
      CALL INVERS(S,SI,ND)
C     **********************************************
```

```
C
C       FINE "UN" AND STORE TO "ODV"
C
C       ************************************************
44      DO 42 I=1,ND
        SUM=0.
        DO 43 J=1,ND
43      SUM=SUM+SI(I,J)*CL(J)
        SN1(I)=SUM
42      CONTINUE
        DO 45 K=1,8
        SUM=0.
        DO 46 L=1,ND
46      SUM=SUM+ODV(K,L)*SN1(L)
        DSC(K)=DT(K)-SUM
45      CONTINUE
        DO 47 I=1,8
        SUM=0.
        DO 48 J=1,8
48      SUM=SUM+WI(I,J)*DSC(J)
        OD(I)=SUM
47      CONTINUE
        SUM=0.
        DO 49 I=1,8
49      SUM=SUM+OD(I)*OD(I)
        ALN=1./SQRT(SUM)
        DO 50 I=1,8
        OD(I)=OD(I)*ALN
        ODV(I,NE)=OD(I)
50      CONTINUE
C       ************************************************
C
C       FINE "RN"
C
C       ************************************************
        CALL RNV(W,OD,DT,NE)
40      CONTINUE
        INDEX=401
        DO 55 I=1,8
        DO 56 J=1,5
        A(J)=ODV(I,J)
56      CONTINUE
        WRITE(5,156)(A(K),K=1,5)
156     FORMAT(5(5X,F15.6))
        WRITE(1'INDEX)A
55      CONTINUE
        CALL BELL
        CALL EXIT
        END
        SUBROUTINE RNV(W,OD,DT,NE)
        REAL OD(8),DT(8),W(8,8),T(8)
        SUM=0.
        DO 1 I=1,8
1       SUM=SUM+OD(I)*DT(I)
        DNM=SUM*SUM
        DO 2 I=1,8
        SUM=0.
        DO 3 J=1,8
3       SUM=SUM+W(I,J)*OD(J)
2       T(I)=SUM
        SUM=0.
        DO 4 K=1,8
4       SUM=SUM+OD(K)*T(K)
```

```
      RN=DNM/SUM
      WRITE(5,140)NE,RN
140   FORMAT(10X,'RN(',I2,1X,')=',F15.6)
      RETURN
      END
      SUBROUTINE INVERS(S,SI,ND)
      REAL S(8,8),SI(8,8)
      COMMON SD(8,8),CM1(8),CM2(8),Y(8),CN,N1,N
      SI(1,1)=1./S(1,1)
      DO 2 N=2,ND
      N1=N-1
      DO 1 II=1,N1
1     Y(II)=S(N,II)
      CALL STY(SI)
      CALL YTTS(SI)
      CALL COST(S)
      CALL LEFTUP(SI)
      DO 3 K=1,N1
      DO 3 L=1,N1
3     SI(K,L)=SD(K,L)/CN
      DO 4 I=1,N1
      SI(I,N)=0.-(CM1(I)/CN)
      SI(N,I)=0.-(CM2(I)/CN)
4     CONTINUE
      SI(N,N)=1./CN
2     CONTINUE
      CALL BELL
      RETURN
      END
      SUBROUTINE STY(SI)
      REAL SI(8,8)
      COMMON SD(8,8),CM1(8),CM2(8),Y(8),CN,N1,N
      DO 1 I=1,N1
      SUM=0.
      DO 2 J=1,N1
2     SUM=SUM+SI(I,J)*Y(J)
1     CM1(I)=SUM
      RETURN
      END
      SUBROUTINE YTTS(SI)
      REAL SI(8,8)
      COMMON SD(8,8),CM1(8),CM2(8),Y(8),CN,N1,N
      DO 1 I=1,N1
      SUM=0.
      DO 2 J=1,N1
      SUM=SUM+Y(J)*SI(J,I)
2     CONTINUE
1     CM2(I)=SUM
      RETURN
      END
      SUBROUTINE COST(S)
      REAL S(8,8)
      COMMON SD(8,8),CM1(8),CM2(8),Y(8),CN,N1,N
      CN=S(N,N)
      DO 1 I=1,N1
1     CN=CN-Y(I)*CM1(I)
      RETURN
      END
      SUBROUTINE LEFTUP(SI)
      REAL SI(8,8)
      COMMON SD(8,8),CM1(8),CM2(8),Y(8),CN,N1,N
```

```
      DO 1 I=1,N1
      DO 1 J=1,N1
1     SD(I,J)=CM1(I)*CM2(J)+SI(I,J)*CN
      RETURN
      END
```

```
C
C     FILE: COMP[50,50]
C
C     ***************************************************
C
C     TRANSFORM THE ORIGINAL EIGHT-DIMENSION FEATURE SPACE
C     TO N (N LESS-EQUAL 3) DIMENSONAL FEATURE SPACE.
C
C     ***************************************************
      REAL A(8,5),X(8,64),Y(3,64),F(64),E(25)
      WRITE(6,147)
147   FORMAT(1X,'THE DIMENSION OF COMPRESSION = ?(MAX.=3)'/'0')
      READ(6,159)NP
159   FORMAT(I2)
      DEFINE FILE 1(408,50,U,INDEX)
      INDEX=401
      DO 11 I=1,8
      READ(1'INDEX)E
      DO 11 J=1,NP
      A(I,J)=E(J)
11    CONTINUE
      END FILE 1
      DEFINE FILE 2(864,128,U,INDEX)
      DO 101 I=1,64
101   F(I)=0.
      INDEX=521
      DO 102 J=1,192
102   WRITE(2'INDEX)F
      DO 1 I=2,63
      INDEX=(I-1)*8+1
      DO 2 J=1,8
      READ(2'INDEX)F
      DO 3 K=1,64
3     X(J,K)=F(K)
2     CONTINUE
      DO 111 II=1,NP
      DO 111 JJ=1,64
      SUM=0.
      DO 112 KK=1,8
112   SUM=SUM-A(KK,II)*X(KK,JJ)
      Y(II,JJ)=SUM
111   CONTINUE
      INDEX=(I-1)*3+521
      DO 5 II=1,NP
      DO 6 JJ=1,64
6     F(JJ)=Y(II,JJ)
      WRITE(2'INDEX)F
5     CONTINUE
1     CONTINUE
      CALL BELL
      CALL EXIT
      END
```

```
C
C        FILE: CL2[50,50]
C
C        ******************************************************
C
C        CLASSIFICATION (SEGMENTATION)
C
C        ******************************************************
         REAL F(64),Y(3,64),X(2,3,225),U(2,3),COV(2,3,3),COVI(2,3,3)
         REAL CV(2),E(3,6),SCOV(3,3),SCOVI(3,3),T(3),EM(3,6),XM(2,3)
         REAL F1(64),SH(64),CA(5,64),TM(2,3),XVX(2),DA(64)
         INTEGER IX(2),IY(2)
C        ******************************************************
C
C        DETERMINE THE ORIGINAL POINTS OF LEARNING SAMPLES
C        , THE COMPRESSION DIMENSIONS AND THE BOUNDARY OF IMAGE.
C
C        ******************************************************
         DEFINE FILE 2(864,128,U,INDEX)
         WRITE(6,10)
10       FORMAT(1X,'READ IX1,IY1,IX2,IY2,ND'/'0')
         READ(6,11)IX(1),IY(1),IX(2),IY(2),ND
11       FORMAT(5I5)
         IXI1=496
         IYI1=552
         IXF1=756
         IYF1=292
         IXI2=504
         IYI2=544
         IXF2=748
         IYF2=300
C        ************************************************
C
C        CHOOSE LEARNING SAMPLES
C
C        ************************************************
         DO 1 I=1,2
         I1=(IY(I)-1)*3+520
         IX1=IX(I)-1
         DO 2 J=1,15
         KK=(J-1)*15
         J1=(J-1)*3
         DO 3 K=1,ND
         INDEX=I1+J1+K
         READ(2'INDEX)F
         DO 3 L=1,15
         KN=KK+L
3        X(I,K,KN)=F(IX1+L)
2        CONTINUE
1        CONTINUE
C        ******************************************************
C
C        CALCULATE MEAN , COVARIANCE MATRIX AND ITS INVERS MATIX.
C
C        ******************************************************
         DO 21 I=1,2
         DO 20 M=1,ND
20       U(I,M)=0.
         DO 22 J=1,ND
         DO 22 K=1,225
22       U(I,J)=U(I,J)+X(I,J,K)
```

```
      DO 29 M=1,ND
29    U(I,M)=U(I,M)/225.
21    CONTINUE
      DO 28 I=1,2
      DO 24 J=1,225
      DO 25 K=1,ND
25    T(K)=X(I,K,J)-U(I,K)
      DO 26 L=1,ND
      DO 26 M=1,ND
26    COV(I,L,M)=COV(I,L,M)+T(L)*T(M)
24    CONTINUE
      DO 27 N=1,ND
      DO 27 N1=1,ND
      CN=COV(I,N,N1)/225
      COV(I,N,N1)=CN
      SCOV(N,N1)=CN
27    CONTINUE
      CALL INVERS(SCOV,SCOVI,ND)
      DO 23 N=1,ND
      DO 23 N1=1,ND
23    COVI(I,N,N1)=SCOVI(N,N1)
28    CONTINUE
C     **************************************************
C
C     FIND THE DETERMINE OF COVARIANCE MATRIX.
C
C     **************************************************
      IF(ND.EQ.3)GO TO 43
      DO 41 I=1,2
41    CV(I)=COV(I,1,1)*COV(I,2,2)-COV(I,1,2)*COV(I,2,1)
      GO TO 49
43    DO 44 II=1,2
      DO 45 KK=1,2
      DO 45 JJ=1,3
      IK=(KK-1)*3+JJ
      DO 45 J=1,3
45    E(J,IK)=COV(II,J,JJ)
      C1=0.
      C2=0.
      DO 46 KK=1,3
      K2=KK+1
      K3=KK+2
      C1=C1+(E(1,KK)*E(2,K2))*F(3,K3)
      C2=C2+(E(3,KK)*E(2,K2))*E(1,K3)
46    CONTINUE
      CV(II)=ABS(C1-C2)
44    CONTINUE
49    VCV=CV(1)/CV(2)
      ALV=ALOG(VCV)
C     **************************************************
C
C     CLASSIFICATION, DISPLAY AND STORAGE
C
C     **************************************************
      CALL VWINDO(1.,64.,1.,64.)
      CALL SWINDO(500,256,300,256)
      DO 51 I=2,63
      INDEX=(I-1)*3+521
      DO 52 J=1,ND
      READ(2'INDEX)F
      DO 50 K=1,64
50    Y(J,K)=F(K)
52    CONTINUE
```

```
          DO 53 M=2,63
          DO 54 N=1,2
          DO 54 J=1,ND
54        XM(N,J)=Y(J,M)-U(N,J)
          DO 55 L=1,2
          DO 55 J=1,ND
          SUM=0.
          DO 56 K=1,ND
56        SUM=SUM+COVI(L,J,K)*XM(L,K)
          TM(L,J)=SUM
55        CONTINUE
          DO 57 L=1,2
          SUM=0.
          DO 58 J=1,ND
58        SUM=SUM+XM(L,J)*TM(L,J)
          XVX(L)=SUM
57        CONTINUE
          XM12=XVX(1)-XVX(2)
          HX=(XM12+ALV)/2.
          IF(HX.GT.0.)GO TO 60
          DA(M)=1.
          GO TO 53
60        DA(M)=0.
53        CONTINUE
          YY=64.-FLOAT(I)
          DO 59 J=1,64
          IF(DA(J).EQ.0.)GO TO 59
          XX=FLOAT(J)
          CALL POINTA(XX,YY)
59        CONTINUE
          INDEX=I+720
          WRITE(2'INDEX)DA
51        CONTINUE
          CALL WINDO(IXI1,IYI1,IXF1,IYF1)
          CALL BELL
C         ******************************************
C
C         ITERATION
C
C         ******************************************
          READ(6,499)IE
499       FORMAT(I2)
          CALL NEWPAG
          DO 500 MI=1,16
          WRITE(6,501)MI
501       FORMAT(10X,'ITERATION',I5)
          INDEX=722
          DO 201 I=1,5
          READ(2'INDEX)F
          DO 202 J=1,64
202       CA(I,J)=F(J)
201       CONTINUE
          DO 251 I=4,61
          INDEX=(I-1)*3+521
          DO 252 J=1,ND
          READ(2'INDEX)F
          DO 250 K=1,64
250       Y(J,K)=F(K)
252       CONTINUE
          DO 253 M=4,61
          P1=0.
          MI=M-3
          DO 353 MM=1,5
```

```
      DO 353 NN=1,5
      NX=M1+NN
353   P1=P1+CA(MM,NX)
      P1=P1-CA(3,M)
      P2=24.-P1
      IF(P1.EQ.24.)GO TO 354
      IF(P1.EQ.0.)GO TO 355
      ALP=ALOG(P1/P2)
      GO TO 356
354   ALP=100.
      GO TO 356
355   ALP=-100.
356   DO 254 N=1,2
      DO 254 J=1,ND
254   XM(N,J)=Y(J,M)-U(N,J)
      DO 255 L=1,2
      DO 255 J=1,ND
      SUM=0.
      DO 256 K=1,ND
256   SUM=SUM+COVI(L,J,K)*XM(L,K)
      TM(L,J)=SUM
255   CONTINUE
      DO 257 L=1,2
      SUM=0.
      DO 258 J=1,ND
258   SUM=SUM+XM(L,J)*TM(L,J)
      XVX(L)=SUM
257   CONTINUE
      XM12=XVX(1)-XVX(2)
      HX=(XM12+ALV)/2.
      ALP=ALP*4.
      IF(HX.GT.ALP)GO TO 260
      DA(M)=1.
      GO TO 253
260   DA(M)=0.
253   CONTINUE
      YY=64.-FLOAT(I)
      DO 259 J=1,64
      IF(DA(J).EQ.0.)GO TO 259
      XX=FLOAT(J)
      CALL POINTA(XX,YY)
259   CONTINUE
      INDEX=I+800
      WRITE(2'INDEX)DA
      DO 359 J=1,4
      J1=J+1
      DO 359 K=1,64
359   CA(J,K)=CA(J1,K)
      INDEX=(I+3)+720
      READ(2'INDEX)F
      DO 360 K=1,64
360   CA(5,K)=F(K)
251   CONTINUE
      DO 261 I=4,61
      INDEX=I+720
      READ(2'INDEX)F1
      INDEX=I+800
      READ(2'INDEX)F
      DO 263 MF=4,61
263   F1(MF)=F(MF)
      INDEX=I+720
      WRITE(2'INDEX)F1
261   CONTINUE
```

```
      CALL WINDO(IXI2,IYI2,IXF2,IYF2)
      CALL BELL
      READ(6,262)IR
262   FORMAT(I2)
      CALL NEWPAG
500   CONTINUE
      CALL BELL
      CALL EXIT
      END
      SUBROUTINE WINDO(IXI,IYI,IXF,IYF)
      CALL MOVABS(IXI,IYI)
      CALL DRWABS(IXI,IYF)
      CALL DRWABS(IXF,IYF)
      CALL DRWABS(IXF,IYI)
      CALL DRWABS(IXI,IYI)
      CALL BELL
      RETURN
      END
      SUBROUTINE INVERS(S,SI,ND)
      REAL S(3,3),SI(3,3)
      COMMON SD(3,3),CM1(3),CM2(3),Y(3),CN,N1,N
      SI(1,1)=1./S(1,1)
      DO 2 N=2,ND
      N1=N-1
      DO 1 II=1,N1
1     Y(II)=S(N,II)
      CALL STY(SI)
      CALL YTTS(SI)
      CALL COST(S)
      CALL LEFTUP(SI)
      DO 3 K=1,N1
      DO 3 L=1,N1
3     SI(K,L)=SD(K,L)/CN
      DO 4 I=1,N1
      SI(I,N)=0.-(CM1(I)/CN)
      SI(N,I)=0.-(CM2(I)/CN)
4     CONTINUE
      SI(N,N)=1./CN
2     CONTINUE
      CALL BELL
      RETURN
      END
      SUBROUTINE STY(SI)
      REAL SI(3,3)
      COMMON SD(3,3),CM1(3),CM2(3),Y(3),CN,N1,N
      DO 1 I=1,N1
      SUM=0
      DO 2 J=1,N1
2     SUM=SUM+SI(I,J)*Y(J)
1     CM1(I)=SUM
      RETURN
      END
      SUBROUTINE YTTS(SI)
      REAL SI(3,3)
      COMMON SD(3,3),CM1(3),CM2(3),Y(3),CN,N1,N
      DO 1 I=1,N1
      SUM=0.
      DO 2 J=1,N1
      SUM=SUM+Y(J)*SI(J,I)
2     CONTINUE
1     CM2(I)=SUM
      RETURN
```

```
      SUBROUTINE COST(S)
      REAL S(3,3)
      COMMON SD(3,3),CM1(3),CM2(3),Y(3),CN,N1,N
      CN=S(N,N)
      DO 1 I=1,N1
1     CN=CN-Y(I)*CM1(I)
      RETURN
      END
      SUBROUTINE LEFTUP(SI)
      REAL SI(3,3)
      COMMON SD(3,3),CM1(3),CM2(3),Y(3),CN,N1,N
      DO 1 I=1,N1
      DO 1 J=1,N1
1     SD(I,J)=CM1(I)*CM2(J)+SI(I,J)*CN
      RETURN
      END




C
C     FILE: PRIKL50,503
C
C     ***************************************************************
C
C     PRINT OUT THE RESULT
C
C     ***************************************************************
      REAL F(64)
      INTEGER IA(64)
      DEFINE FILE 2(864,128,U,INDEX)
      WRITE(6,10)
10    FORMAT(1X,'FILE 721-784 ,NF=1'/1X,'FILE 801-864,NF=2'/)
      READ(6,11)NF
11    FORMAT(I2)
      IF(NF.EQ.2)GO TO 22
      INDEX=724
      GO TO 33
22    INDEX=804
33    DO 1 I=4,61
      READ(2'INDEX)F
      DO 2 J=4,61
2     IA(J)=INT(F(J))
      WRITE(5,44)(IA(II),II=4,61)
44    FORMAT(10X,58I1)
1     CONTINUE
      CALL BELL
      CALL EXIT
      END
```

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A098285
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle) Note on Statistical Texture Discrimination	5. TYPE OF REPORT & PERIOD COVERED Technical Report
	6. PERFORMING ORG. REPORT NUMBER SMU-EE-TR-81-9
7. AUTHOR(s) C. H. Chen Rong-Hwang Wu	8. CONTRACT OR GRANT NUMBER(s) N00014-79-C-0494
9. PERFORMING ORGANIZATION NAME AND ADDRESS Electrical Engineering Department Southeastern Massachusetts University North Dartmouth, Mass. 02747	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NR 042-422
11. CONTROLLING OFFICE NAME AND ADDRESS Statistics and Probability Program Office of Naval Research, Code 436 Arlington, Virginia 22217	12. REPORT DATE April 24, 1981
	13. NUMBER OF PAGES 25
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report) Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Texture features; Image feature extraction.
Optimum discriminant vector; Feature space transformation;
Bayes classification rule; a priori probability.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Same as Summary

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)